Author(s): Melomey, Divina., Williams, Godfried., Imafidon, Chris., Perryman, Roy.
Article Title: Modelling Mobile Agent Mobility in Virtual Learning Environment (VLE) using Fitness function.
Year of publication: 2008
Citation: Melomey, D. et al (2008), 'Modelling Mobile Agent Mobility in Virtual Learning Environment (VLE) using Fitness function.' in Rienties, B., Giesbers, B., & Gijselaers, W.H. (2008). Student Mobility and ICT: Can elearning overcome barriers of life-long learning? Maastricht: FEBA ERD Press. pp 159-165
Link to published version:
http://www.fdewb.unimaas.nl/educ_v2/MASTER/Documents/Proceedings_S_ICT2008_Final.pdf
ISBN: 978-90-813727-1-8
DOI: (not stated)

Publisher statement:
http://www.fdewb.unimaas.nl/educ_v2/default.asp

Modelling Mobile Agent Mobility in Virtual Learning Environment (VLE) using Fitness function

Divina Melomey, Godfried Williams, Chris Imafidon, Roy Perryman
University of East London, School of Computing and Technology
Docklands Campus, United Kingdom,
{divina,G.Williams,chris12, perryman }@uel.ac.uk

Abstract: Virtual learning Environments are driven by distributed systems. Effective distributed systems communications requires intelligent mobility as a vehicle to enabling seamless resource sharing and access to services. The nature of VLEs requires software tools for managing and making learning enjoyable and less painstaking. Mobile agents enable different software and services to collaborate in information sharing, adapt to new service requirements, demonstrate cooperation in a system environment, however being independent and autonomous. These requirements are essential in achieving mobility in VLEs. This work presents a novel fitness function as a key feature of a generic software methodology for modeling mobile agent mobility in VLEs.

Introduction

Virtual Learning Environments (VLEs) are group of software used for managing and enhancing learning electronically (Roach & Stiles, 1998). This facilities and functionality enables tutors, instructors and students communicate online (Ginsburg, 1998). VLE should have the capabilities to enhance student learning experience based on the requirements of the programme a student is enrolled hence enriching students learning experience. Heaton-Shrestha, , Ediringha, Burke and Linsey (2005) also defined VLE as web-based software products providing sets of internet tools to enable teaching materials to be managed. Pablo and Wallace (2001) explained the VLE is not only dependent on the its accessibility, availability and the integration of the technology for the benefits of students but rather on the willingness of tutors to embrace and use computers for the delivery of course materials. Apart from VLE supporting teaching, learning and certain administrative functions, it also has the ability to facilitate communications among learners (Booth & Hulten 2003). The modes of communication are both asynchronous and synchronous. Again, VLE mode of delivery can be synchronous, asynchronous and/or both (Chen, Li & Shyu, 2003). These forms of communications are emails, booking appointments, negotiating assignment deadlines, social interactions with other students via blackboard learning.

This paper reports on a study conducted to ascertain the requirement for developing Virtual Learning Environments (VLEs) and how these needs are met using fitness function for modelling the solution to meet the requirements and demands of such as system. The systems used for this study was University of East London blackboard Learning System called UEL Plus. We realized that UEL Plus has multiple features to support teaching and learning. UEL Plus provides an improved communication, access to resources and advanced assessment capabilities. Our study focused fundamentally on the UEL Plus which part of VLE. The rest of the paper is organized as follows: Section 2 will describe end user categories and section 3 will highlight the mobile agents as a solution. Section 4 will introduce mobile agent fitness function and Section 5 will discusses the mobility in VLE in section 6 draws conclusions.

User Categories

We identified two main user groups for this study. They are front-end and back-end users. According to Sampson, Karagiannidis, Schenone and Cardinali (2002) formal, vocational, life long and occasional learners fall under the front end users category while individuals, software houses and other organization whose main interest are developing management learning and virtual learning software. Basic functions and or task on a VLEs are;

1. Authentication and authorization
2. Editing and saving personal settings
3. Navigation through the site

4. Using available communication tools
5. Building course content
6. Assessment
7. File Upload

Front end users need these basic functions to be user friendly and easily accessible. This form of interactions between users and the software is at the heart of e-learning development.

Back end users uses information and input from the front end users to map up these functions of the front end users to the solution provided by back end users with respect to developing knowledge repositories and resources.

Mobile Agents an Alternative Solution

Our experiences in evaluating UEL Plus identified certain areas where an agent could be used in modeling interactions and communications during the systems development as we believe that it will considerably improve performance and front end user experiences of UEL Plus. The areas where we had feedback relating to front end user experiences were:

- Uploading of files
- Maintaining files and folders on VLE
- Using communication tools for creating asynchronous discussion, emails and chat
- Monitoring and tracking progress of students
- Other emerging technologies that could be added on

Based on these feedbacks, we proposed a solution into the modelling of mobility in mobile agent for VLEs. Gutl et al. (2004) identified three main objectives as an innovative solution in e-learning systems. These objectives were;

- Personalized retrieval of information,
- Presentation and management of relevant learning material in a timely fashion; ability to support teaching and learning paradigms and lastly
- An improvement on knowledge with respect to front end users behaviour in human to computer interaction.

In the following section we will show how we used the fitness function to model solutions for the critical areas of applications that require mobility such as VLEs.

MOBILITY FITNESS FUNCTION

Mobility fitness function is a function derived from an algorithm, based on the concept of survival of the fittest in genetics. In this section we defined elements for the mobility requirement for the mobile agent. The list is not exhaustive but only a representation for the fitness function.

Let F be the function denoting key mobility requirements for a mobile agent.
Let f_1 to f_{15} be elements in the same set F.

f_1 = Synchronization
f_2 = Latency
f_3 = Abstraction
f_4 = Polymorphism
f_5 = Inheritance
f_6 = Persistency
f_7 = Calling
f_8 = Invocation
f_9 = Message Passing
f_{10} = Naming

f_{11} = Addressing
f_{12} = Encoding
f_{13} = Availability
f_{14} = Replication
f_{15} = Self Protective and Certified

Melomey, Williams, Imafidon, & Perryman (2008) established the implementation of generic mobility fitness function based on the following steps:

- Initial population should be randomly created for mobile agent *m(0)*
- the fitness function *U(m)* should be computed for each individual mobile agent *m* in current population *m(t)*
- probability for selection *p(m)* for each individual mobile agent in *m(t)* should be defined, such that the probability *p(m)* is equal to *U(m)*
- *m (t + 1)* generated
- Selection of individual mobile agents using probability *m(t)* to produce new agents which is known as offspring via crossover, mutation or reproduction

Let $F(X) = (x_1 \ldots x_n)$

The fitness function $U(m) = (x_1 \ldots x_n)$

Where $U(m) = (1/e+x)^2$

$$U(m) = m(t) + \sum_{x=1}^{n} m(t+1)\, f(x)$$

The above expression represents a fitness function in an inverse relationship to a fitness solution.

The fitness solution derived from the fitness function is applied in the second of the four major phases thus;

1. Mobility requirement
2. Mobility analysis
3. Mobility design and
4. Implementation of code.

Fitness Function for VLE

In the following subsection we will show how we used our fitness function to provide solution for VLE issues identified in section 2.

Addressing

There are certain elements that need to be present for an entity say agent to be able to travel from its platform of origin Hp_i to a host platform Vp_n. These requirements are required to perform address resolution prior to process migration. Three elements that need to be present are:

Receiver identification (RID)
Packet identification (PID)
Transmission Frequency of physical layer (TF)

Let R be the set requirement RID, PID, TF

Let H be the set header fields that contains control information
L be length of the packet
p be payload type
s be sequence numbers
i be integrity check information

$R \subseteq H$

Each computing platform is identified by global assigned address. A process will be able to migrate if it contains a header field carrying control information. The address resolution client which is the host platform needs to verify the integrity, authenticity and the logical address for resolving information sent across different platforms.

A platform hosting each mobile agent need to ensure mobile agents on its platform has a valid server and address resolution is also valid. Authorisation of available address to be used should be authorised by servers in order to ensure validity of the address.

Replication

High availability of services is paramount to mobile distributed computing as it enhances performance. It is a technique that is used to maintain copies of data in geographically dispersed environment and also as a back up in the event of loss of data or a systems failure (Coulouris, Dollimore, & Kindberg, 2005). The fitness of a replica will be measured in real time by the function of the differences in elapsed time. This ensures consistency and correctness at anytime for events. This represented as:

$F(t): f_{t+1} - f_{t-1}$

Where f_{t+1} is the current time replica server was accessed

f_{t-1} last known time a replica was accessed

Remote Method Invocation (RMI)

A method transparently invoked from process A to process B across a network as if it were a local method is termed as remote method invocation (Coulouris et al.2005; Williams 2000). This holds true for object oriented language rather than a procedural language. Invoking a method remotely involves two processes:

1. a reference to the remote object
2. a registry to store remote references

Let n be the number of identified elements for solution X

x_i be elements in X

$f(x_i)$ the fitness of x_i

The fitness of F can then be defined as

$$F(X) = {}^{1}/_{n} \sum_{i=1}^{n} f(x_i);\ n>0$$

We define the average fitness above as average fitness for the elements in the mobility requirements as identified.

$F(x): Hp_i \rightarrow Vp_n$

Persistency

The Object Management Group (OMG) service stipulates a typical structure for persistency. This should consist of persistent ID, persistent object, persistent object manager, persistent data store and protocol. A persistent object or entity that need to travel from Home platform (Hp_1) to visit (n) number of visiting platform (Vpn) require a reference ID, a dynamic state that lives the duration of the process and a persistent state that will be used for reconstruction of the dynamic state in case of a failure. These conditions qualified for an entity to be mobile in an environment.

Naming services

The Sun Microsystem naming services system administration guide defines naming services as a central repository that computers, end users, and applications communicate together across the network. In this work, we also define name services as integrated services that manages all name information and hierarchies and also as an autonomous feature for transparency and persistency of entities (Melomey et al. 2007). Its function is to provide basic function and mapping of name to address on the network. In order to get the remote computer's address, the program must request assistance from say Hp_1 from the domain name services (DNS) database running on that platform. DNS is a naming service which provides identification for computers on the internet . The name server uses Hp_1 as part of the request to find IP address of the remote computer. The name server returns this IP address to the Hp_1 only if the host name is in its database. It uses a logical tree to resolve names as part of the service

Synchronization

Synchronization is important to maintain consistency of processes from H_p to V_{pn} at any given time (Coulouris et al.2005). The concept of clock synchronization deals with the understanding of ordering of events occurrence as produced by current processes. These events occur between message sender and message recipient for example from process A to process B. Clock synchronization is required to provide mechanism that can assign numbers sequentially based on agreement between sending and receiving processes. Several algorithms were developed over past decades. Lamport (1978) introduced the concept of an event happening before another in distributed environment. The notion is illustrated between event *a* and *b*; *a* à *b* where *a* "happens before" *b*. Another algorithm developed by Lamport and Meilliar-Smith (1985) require a reliable connected network to handle fault. Christian's algorithm measures in local time the time at which a message is sent (T_0) and the time at which a message is received (T_1). This is done by issuing a remote procedure call to a time server to obtain the time. The delay in the network is then estimated as $(T_1 - T_0)/2$ (Christian, 1989). Hence the new time can be said to be the time returned by the server and in addition to time elapsed by the server to generate the timestamp. This is expressed by
$\text{Time}_{new} = \text{Time}_{server} + (T_1 - T_0)/2$. There is also the Berkeley algorithm which was developed by Gusella and Zatti (1989). Berkeley algorithm was based on the assumption that any computer on the network has an accurate time which can be used for synchronizing time between processes. This assumption may introduce delays and losses depending on the network and also due to the distributed nature in accessing the network and the processing capabilities on the learning system.

Let S = Synchronization
H_p = visiting platform
V_p = visiting platform
V_{pn} = *n* visiting platform
P_n = *n* number of processes

The timescale for measuring Δs is important where S which synchronisation is a derivative of the f(x) which is $^{\Delta f}/_{\Delta s.}$ Measuring the short time for *n* processes is dependent on how fast changes occur in the system. The time range between which *n* process leaves H_p and arrives at V_{pn} can be expressed as:

$$F(x) \text{ à } \Delta t = \int_{t}^{t+\Delta t} f(s)\, dt \quad \text{where the interval is } [t, t+\Delta t]$$

F(*x*) is a complex system during its evolution; the system may change its own F

Discussion

In our study using UEL Plus, we analyzed feedback, identified student lecturer issues and evaluated mobility solutions for back-end user category. Solutions we designed using mobile agent oriented approach addressed synchronization, remote method invocation, addressing and naming services, persistency and replication of data. We examined the persistency of data and how they were mapped into the objects. We enabled the mobile entity to have an internal mechanism which acts as a persistency layer such that it will encapsulate database access from other objects. In this manner, data persist after any form of interruption and interaction occurs during the course. A fitness function for modelling and testing features appropriate for persistency of objects is critical in such as environment.

Front end users are more interested in up to date, timely and current state of databases. This implies that concurrent data access and update of repositories should be synchronized. This is more crucial when it comes to coursework submission for group projects, where continuous and joints updates are required from individual team members when approaching deadlines. Synchronization then becomes an issue for the back end users to deal with in order to ensure consistency of data, processes and clock synchronization of various remote devices connected to the network infrastructure. Our work indicates that there is a connection between replication of data at various server locations with respect to change in time among primary and secondary servers. This also applies to resolution of names and addresses.

We had the understanding that front-end users were looking for a unified point of authentication for ensuring coherent and an organized teaching and learning resource platform. Consistent and coordinated naming of objects and identification of processes underpins the need for metadata as a means of providing effective mobility. These needs are met based on the conditions that must be met for remote method or data invocation's fitness function criteria. The fitness function measures the suitability for elements mobility in the VLE.

Conclusion

In this paper, we presented an overview of VLE and user categorization. We also presented fitness function for mobility as alternative solution to traditional approaches in eliciting requirements for implementing mobility in VLEs. This mobility fitness function was further illustrated by applying it for mobility element requirements specification. This was further narrowed down to individual mobility requirement mapped unto their fitness solution applicable to the development of VLE and it was used to provide a solution tailored for simulating effective mobility in UEL Plus.

Currently, work is being done to integrate this fitness function as part of a generic methodology for capturing mobility in mobile agent based systems and applications. This when concluded will provide a standard methodology for building applications where mobile agents are seen as an alternative approach to information systems development.

References

Booth, S., & Hulten, M. (2003). Opening dimension of variation: An empirical study of learning in a web-based discussion. Instructional Science Vol.36 (No.1&2), 65-86.

Chen, S.-C., Li, S.-T., & Shyu, M.-L. (2003). Model-Based System Development for Asynchronous Distance Learning. International Journal of Distance Education Technologies, Vol.1 (No. 4), 39-54.

Christian, F. (1989). A Probabilistic Approach to Distributed Clock Synchronization. Distributed Computing Vol. 3, 146-158

Coulouris, G., Dollimore, J., & Kindberg, T. (2005). Distributed Systems, Concepts and Design (4th ed.): Addison-Wesley Publishers.

Ginsburg, L. (1998). In Technology, Basic Skills, and Adult Education: Getting Ready and Moving Forward. Information Series (No. 372).

Gusella, R., Zatti, S. (1989). The Accuracy of ClocK Synchronization Achieved by TEMPO in Berkeley UNIX 4.3BSD.IEEE on Software Engineering, Vl.15 (No.7).

Gütl, C., Pivec, M., Trummer, C., García-Barrios, V. M., Mödritscher, F., (2005). AdeLE (Adaptive e-Learning with Eye-Tracking): Theoretical Background, System Architecture and Application Scenarios. European Journal of Open, Distance and E-Learning (EURODL)(2005/II).

Heaton-Shrestha, C., Ediringha, P., Burke, L., & Linsey, T. (2005). Introducing a VLE into campus-based undergraduate teaching: Staff Perspectives on its impact on teaching. International Journal of Educational Research Vol. 43(6), 670-386.

Lamport, L (1978). Time, Clock and Ordering of Events in Distributed Systems. Communications of ACM Vol21(No. 7).

Lamport, L & Melliar-Smith, P. M.(1985). Synchronizing Clocks in the Presence of Faults. Journal of ACM Vol.32(No. 1), 52-78.

Melomey, D., Williams, G., Imafidon, C., & Perryman, R. (2008). *A Fitness Function for Capturing Mobile Agent Mobility on Games Platform* Paper presented at the 12th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia & Serious Games. , Louisville, Kentucky, USA.

Melomey, D., Williams, G., & Imafidon, C. (2007). *Mobility Requirements on Game Platforms: An Agent Perspective* Paper presented at the 11th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia & Serious Games. , University of La Rochelle, La Rochelle, France.

Object Management Group (2000). Persistence Object Service Stand-alone document.

Pajo, K., & Wallace, C. (2001). Barriers to the uptake of web-based technology by university teachers. Journal of Distance Education Vol. 16(No.1), 70-84.

Roach, M. P., & Stiles, M. J (1998). COSE - A Virtual Learning Environment founded on a Holistic Pedagogic Approach. CTI: Software for Engineering Education (No.14).

Sampson, D., Karagiannidis, C., Andrea, S., & Fabrizio, C. (2002). Knowledge-on-Demand in e-Learning and e-Working Settings. Educational Technology and Society, Vol.5 (No. 2), 107-112.

Sun Microsystem, Inc. (2003). System Administration Guide: Naming and Directory Services (DNS, NIS, LDAP).

Williams, G.B. (2000). Technical Notes in RMI. Archival and Unpublished.